Machine Learning Applications In 5G Network Orchestration

David Friedlein, Philippe Buschmann*
*Chair of Network Architectures and Services
School of Computation, Information and Technology, Technical University of Munich, Germany
Email: david.friedlein@tum.de, phil.buschmann@tum.de

***Abstract*—The ability to virtualize and separate multiple networks on top of a common physical infrastructure allows network providers to serve different needs. With this approach, 5G networks can better support new technologies such as self-driving cars, which is not possible with traditional one-size-fits-all architecture. This is possible while also reducing the cost for the operators. The drawback is that it requires a lot of configuration and management to function optimally. Machine learning is a possible solution to simplify and automate this work.**

This paper analyses and compares multiple different proposed implementations of machine learning in the network slicing process. We see that all approaches provide benefits but they can not be directly compared to each other, because the measurements are too different.

***Index Terms*—5G, network slicing, machine learning, software-defined networks**

1. Introduction

Earlier mobile standards like 4G and 3G are mainly designed for smartphones and thus have a design solely focused on this purpose. With ongoing technological development, new and different use cases arise. These include self-driving cars, telemedicine and Internet of Things (IoT). All use cases for mobile networking can be divided into three classes following specifications from the International Telecommunication Union (ITU):

- **enhanced mobile broadband (eMBB)** Mainly meant for smartphones which need high data rates and a large area covered due to their mobility.
- **massive machine type communication (mMTC)** Sporadical communication of a large number of devices in a small area. It is used for IoT devices like sensors, which only send small amounts of data in large timeframes. Packet loss is not a significant problem.
- **ultra reliable low latency communication (uRLLC)** Communication with access over 99.9999% and end-to-end latency of less than 50 ms is required for some industrial use cases. For example smart connected fabrication plants.

For every class the network has to fulfill different needs. Serving all of these classes with one network while providing a consistent quality of service (QoS) is hard to achieve. A potential solution could be to build multiple different radio access networks, each specialized for one class. However, the development of multiple networks leads to a high amount of additional costs for network providers.

A better solution is network slicing. It allows network providers to use one physical network to serve all traffic classes while still providing a consistent QoS. It works by separating the network into multiple network slices (NS), which are tailored to a specialized purpose.

The management of all these slices can be quite complicated. Machine learning can simplify and automate this management.

The remainder of this paper analyzes three different approaches. Section 2 explains the background of network slicing and the enabling technologies. Section 3 explains three different papers and their results, which are then discussed in Section 4. Section 6 concludes the paper.

2. Background

In this section we explain the key technologies that enable network slicing.

2.1. Network Function Virtualization

Network function virtualization (NFV) is a concept that enables virtualization to separate hardware from functionality [1]. Network functions (NF) like virtual firewalls or virtual load balancers can be deployed on servers to run these network functions on any server. This helps create flexible networks by deploying necessary NFs on servers when needed. It reduces the dependency on special hardware and the placement of servers in the network.

This technology is crucial for network slicing

2.2. Software Defined Networking (SDN)

Software defined networking (SDN) physically separates the network control plane from the forwarding plane [2]. The forwarding plane consists of all the hardware that forwards packets while the control plane consists of one or more SDN controllers. The control plane has knowledge about the whole network and its policies. With this information, it makes all the routing decisions and communicates them via different protocols to the forwarding plane. Then routers and switches follow the decisions of the control plane and forward the packets. In comparison to a classical network where every router makes its own decisions about forwarding packets, SDN centralizes the routing process. These differences allow

 doi: 10.2313/NET-2023-11-1_13

the network a faster adaption to changes in the network or the users' needs.

SDN is necessary for network slicing to adjust the routing decisions to changing network slices and their different routing needs.

2.3. Network Slicing

Using the above explained technologies modularizing networks is possible. The NGMN (Next Generation Mobile Network) has introduced network slicing for 5G in [3]. It allows the creation of multiple logically independent networks that operate on top of a unified physical infrastructure. Network providers can customize these logical networks to provide different services and performance levels to meet the demand of multiple clients at the same time.

For the creation of these slices three required layers were defined by the NGMN [3]:

- The **infrastructure resource layer** comprises all the physical resources, which includes access nodes, cloud nodes, end-user devices such as smartphones and wearables and even the links between these devices. All devices have different capabilities and can fulfill different roles in the network. These capabilities and roles can be controlled and monitored through an application programming interface (API).
- The **business enablement layer** contains all of the functions and configuration parameters of the network devices. Some of these functions offer different levels of performance, which are used to differentiate the network slices. They are separated into modular blocks and can be loaded onto required devices by an API.
- The **business application layer** contains all applications and services provided by the network operator or different enterprises.

The 3 layers are connected by the **E2E management and orchestration entity**. This entity controls the creation, scaling and geographic distribution of resources of all network slices. It defines a slice depending on the use case and applications needed. It chooses the required network functions with specific performance levels to map those onto the device in the infrastructure resource layer. It makes decisions about the scaling for the lifetime of the slice. It shifts resources between slices to optimize performance.

2.4. 3GPP Specification

The 3GPP defines a management and orchestration architecture. The communication service management function (CSMF) translates incoming requests for services into requirements for the network. These requirements are sent to the Network Slice Management Function (NSMF), which chooses or generates the slice blueprint optimal for the requirements. A slice blueprint contains all needed NFs, their connections and configurations. After the slice is instantiated the NSMF manages it until it is decommissioned.

A Network Slice instance (NSI) is a group of NFs. "An NSI is composed of NFs shared between two or more slices, as well as dedicated NFs" [4].

3. Network Slicing with Machine Learning

In recent years a lot of research about the usage of ML for resource orchestration has been conducted. Multiple research groups have proposed different methodologies to include machine learning in the decision process. Some of these are explained in the following.

3.1. Artificial Intelligence for Slice Deployment and Orchestration

Dandachi et al. [4] propose two new approaches for ML based on the 3GPP NSMF architecture. They define a novel architecture that is compatible with the 3GPP design and includes three new functions:

The slice analytics (SA) function minimizes required resources by sharing them between slices. If multiple slices want to use the same NF they can be grouped in a common NSI.

The admission control (AC) decides whether new slices can be created or have to be dropped because of resource shortage.

This function can be combined with the congestion control (CC) function, which scales slices up and down as needed, to build a cross-slice admission and congestion control (CSACC).

3.1.1. Slice Analytics (SA). The two main tasks of the SA are the classification of slices and the reduction of required resources. For this purpose, it receives the slice blueprint and the resource requirements for new slices. If the requested slice requires NFs and resources that are already used by other slices, they can be shared between the two slices. This reduces the number of new resources that have to be allocated, which then reduces the rate of denied requests due to a shortage of available resources.

Depending on the specific slices and services running on them, the amount of NFs that can be shared is different. This allows the classification into elastic and non-elastic slices [5]. The authors of [4] only consider elastic slices in their research. The usage is explained in the setting of a sports event: Different broadcasts will use the same images, which can be shared, but will provide different commentary, which has to be separated.

The performance of two different algorithms for the grouping are analysed. The first algorithm finds an existing NSI with the highest amount of overlapping NFs using the Jaccard similarity. For each new slice, the best group can be calculated and only the missing NFs are created and added to the NSI.

The second algorithm uses spectral clustering [6] to create NSIs that reduce resource usage. This algorithm does not find an existing NSI to which new slices fit but calculates an optimal grouping for all existing and new slices. This has a the higher complexity ($\mathcal{O}(N^3)$) [6] compared to calculating the Jaccard similarity ($\mathcal{O}(N)$) and thus the authors recommend executing this recalculation in larger time intervals or when the system is overloaded.

Dandachi et al. classify all slices into either guaranteed quality-of service (GS) slices, which have a high priority, or best effort (BE) slices, which have a lower priority. Additionally, if the system needs more resources for GS slices these can be taken from BE slices. For each class, a queue exists, in which new slice requests are inserted until they are created. After the new slice request has been grouped with other slices it is inserted into one of two queues depending on the class of the slice.

3.1.2. Cross-slice admission and congestion control (CSACC). The CSACC function decides which queued slices are accepted and how many resources get assigned to the slices. The goal is to maximize the number of accepted slices while reducing the probability that a new slice request has to be dropped because the queue is full. The resources of each slice can not be reduced beyond a minimum level to prevent extreme degradation in the QoS.

For this, reinforcement learning is employed. State-Action-Reward-State-Action (SARSA) aims to find a policy that maps the state of the system to an action, which maximizes the reward. To enhance this model the authors use linear function approximation.

3.1.3. Performance results. Comparing only the slice admission with SARSA to the CSACC with SARSA, the latter method shows an improvement with up to 23% reduction in dropped slice requests. However, this improvement is only possible if new GS slices are requested with a probability of less than 70%. Higher values show no difference between the two methods.

Using the SA function, the rate of dropped slices is even further reduced. Up to 44% reduced drop rate is achieved by using CSACC with SARSA and SA with spectral clustering compared to not using any function, as seen in Figure 1.

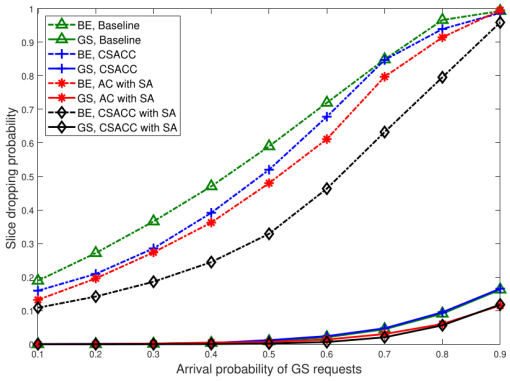

Figure 1: Slice dropping rate as a function of arrival probability from [4]

3.2. Artificial Intelligence for Elastic Management

In a paper from Gutierrez-Estevez et al. [7] the concept of resource elasticity (which was defined in an earlier paper from the same group) is being used as the basis of multiple ML approaches. Resource elasticity describes the ability of a network to automatically and smoothly adapt to changes in the system. This elasticity can be applied to three areas:

- **computational elasticity** in the operation of VNFs
- **orchestration-driven elasticity** in the placement of VNFs
- **slice-aware elasticity** in the distribution of resources between slices

3.2.1. Computationally Elastic Scheduler. One of the more computationally expensive NF is the media access control (MAC) scheduler. It is responsible for assigning bandwidth resources to different devices in a network and deciding on which modulation and coding scheme (MCS) to use. Depending on the signal-to-noise ratio (SNR) in the connection, different MCSs are best suited and have different computational complexities. Contextual bandits are an ML approach that tests different randomized policies. The policies are finetuned regarding environmental conditions. The predicted SNR for a given user is a necessary condition for the contextual bandits. Applying a long short term memory network to this predicts the SNR.

3.2.2. Slice-Aware Resource Management. The authors of [7] design algorithms, which are supposed to optimally allocate/de-allocate resources to individual slices. They have to consider QoS requirements, Service Level Agreements (SLAs) and demands of the slices. These algorithms can be applied to different problems.

The authors use a deep neural network to forecast the amount of traffic in the future. This helps to allocate additional resources, if a large group of users, which increases the demand, is predicted, or de-allocate resources at day times when little to no traffic is expected. Algorithms to adjust different settings in the network can also be improved with this data.

Predicting the movement of people helps to adjust the settings of cell towers such as the beam pattern or even allocate towers at different locations to the slice. Identifying groups of people and predicting their movement can be done by ML algorithms. The position and demand of users are necessary to guarantee reliable coverage.

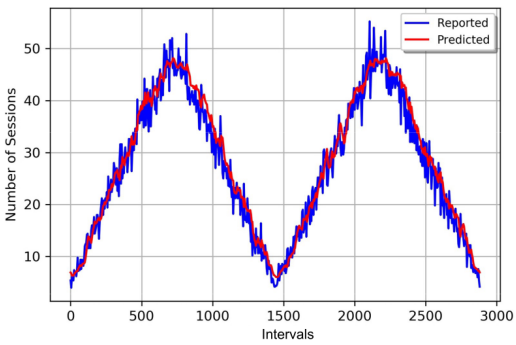

Figure 2: Traffic prediction results from [7]

3.3. Machine Learning Based Resource Orchestration for 5G Network Slices

The group of Salhab et al. [8] proposes a novel architecture that includes ML in the management and

 doi: 10.2313/NET-2023-11-1_13

orchestration process.

3.3.1. System design. The proposed network architecture is not based on the 3GPP architecture but contains four other components as seen in Figure 3. The first part is the **gatekeeper**. To aggregate and sort traffic into the correct slices, first a marking and classification phase is needed. Using the slice blueprints and tenant requests, the gatekeeper generates requirements using supervised learning. This allows us to choose the correct blueprint for each requested slice.

These policies are then handed to the **decision maker**, which is composed of a **forecast aware slicer** and an **admission controller**. The forecast aware slicer uses regression trees to predict the required ratio of all network slices. It achieves this using different information about the traffic. Using this and the current load on the network, the admission controller decides whether to grant requests for new slices or not.

If the request is accepted it gets sent to the **slice scheduler**. Its purpose is to find a schedule that serves all slices and minimizes the total time needed. Salhab et al. prove this to be an NP-hard problem [9] and provide a heuristic for solving it.

Denied requests are sent to the **resource manager**, which uses micro-services to automatically scale resources. If additional resources are needed to allow a new slice to be accepted it reduces the resources for other slices. Available resources can be assigned to slices, increasing their performance. The authors use Reinforcement learning to optimize decision making and improve the system's utilization.

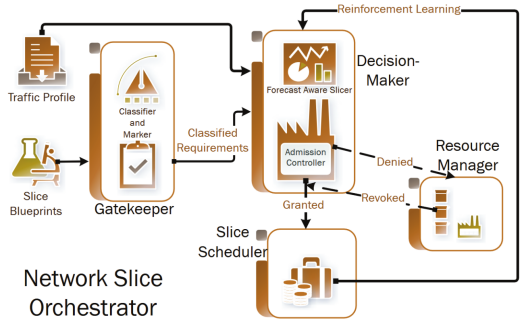

Figure 3: Block diagram of the architecture from [8]

3.3.2. Performance. The paper compares different machine learning algorithms running on the above-mentioned architecture. The first benchmark compares different models used for the classification. Most of the models achieve a prediction accuracy of over 90%. The highest accuracy of 98% was produced using a linear discriminant model.

The second benchmark compares different models used to predict the optimal slice ratios. Different tree algorithms (simple trees, medium trees and complex trees) achieved the lowest root mean squared error (around 5%) and the highest prediction speeds. These results, however, come at the expense of a longer training time compared to linear models. Since the models' training is infrequent, this tradeoff can be accepted. To validate the usefulness of the tree models, the authors compare them to the theoretical optimum, a static slice ratio and a random slice ratio. The ML model performed the best, with an average 5% gap to the theoretical optimum. The random approach performed the worst with a 30% gap.

For the last test, the setup was run with and without traffic forecasting. Using the forecasting the throughput of the system increased by approximately 30%.

4. Evaluation

All approaches are shown to be beneficial in some aspect. But comparing them against each other is difficult because different papers use ML to improve other aspects of the system. Moreover, all of the approaches analyzed in this paper measured different metrics of the network or the ML models. The authors of [7] only show the accuracy of the ML models, but not any performance results obtained from implementing them. Paper [4] focuses on the probability, that a new slice request has to be dropped. The last paper [8] includes the ML accuracy and the throughput of the system with the implemented methods.

Some aspects of an approach can not be measured with numbers. For example, the system architecture in [4] is based on the 3GPP architecture and the system in [8] introduces a completely new architecture. Both provide certain advantages and disadvantages. A standardized system makes it easier to expand and compare to other systems on the same architecture. Creating a new architecture allows the system to be better specialized for a certain use case.

Some current problems in this research area include the lack of data for training supervised models. Because 5G is not widely deployed and the hardware is expensive, collecting real-world data for training is difficult.

5. Related work

There have been multiple other surveys about this research area, which focus on certain types of approaches. Some focus on the applications of 5G and ML for IoT devices Wijethilaka et al. [10], Khan et al. [11]. Others concentrate on deep reinforcement learning Hurtado Sánchez et al. [12]. The survey from Su et al. [13] concentrates on mathematical models.

6. Conclusion and future work

In this paper, we examine different approaches to utilizing ML in the management and orchestration process for network slicing.

We found that ML seems to improve many aspects of the management and orchestration process. One architecture has been shown to increase the throughput of the system [8] other designs increase the number of slices that can be run on a system.

A possible solution to better compare different approaches would be to implement different designs on the same setup and measure the same parameters. This could be done in future work.

References

[1] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network Function Virtualization: State-of-the-Art and Research Challenges," vol. 18, no. 1, 2016, pp. 236–262.

[2] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A Survey on Software-Defined Networking," vol. 17, no. 1, 2015, pp. 27–51.

[3] "NGMN 5G white paper," 2015.

[4] G. Dandachi, A. De Domenico, D. T. Hoang, and D. Niyato, "An Artificial Intelligence Framework for Slice Deployment and Orchestration in 5G Networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 2, pp. 858–871, 2020.

[5] D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, K. Samdanis, and X. Costa-Perez, "Optimising 5G infrastructure markets: The business of network slicing," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, 2017, pp. 1–9.

[6] S. Tsironis, T. Mauro Sozio, and M. Vazirgiannis, "Accurate Spectral Clustering for Community Detection in MapReduce," 2013.

[7] D. M. Gutierrez-Estevez, M. Gramaglia, A. D. Domenico, G. Dandachi, S. Khatibi, D. Tsolkas, I. Balan, A. Garcia-Saavedra, U. Elzur, and Y. Wang, "Artificial Intelligence for Elastic Management and Orchestration of 5G Networks," *IEEE Wireless Communications*, vol. 26, no. 5, pp. 134–141, 2019.

[8] N. Salhab, R. Rahim, R. Langar, and R. Boutaba, "Machine Learning Based Resource Orchestration for 5G Network Slices," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[9] N. Salhab, R. Rahim, and R. Langar, "Throughput-Aware RRHs Clustering in Cloud Radio Access Networks," in *2018 Global Information Infrastructure and Networking Symposium (GIIS)*, 2018, pp. 1–5.

[10] S. Wijethilaka and M. Liyanage, "Survey on Network Slicing for Internet of Things Realization in 5G Networks," vol. 23, no. 2, 2021, pp. 957–994.

[11] L. U. Khan, I. Yaqoob, N. H. Tran, Z. Han, and C. S. Hong, "Network Slicing: Recent Advances, Taxonomy, Requirements, and Open Research Challenges," vol. 8, 2020, pp. 36 009–36 028.

[12] J. A. Hurtado Sánchez, K. Casilimas, and O. M. Caicedo Rendon, "Deep Reinforcement Learning for Resource Management on Network Slicing: A Survey," *Sensors*, vol. 22, no. 8, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/8/3031

[13] R. Su, D. Zhang, R. Venkatesan, Z. Gong, C. Li, F. Ding, F. Jiang, and Z. Zhu, "Resource Allocation for Network Slicing in 5G Telecommunication Networks: A Survey of Principles and Models," *IEEE Network*, vol. 33, no. 6, pp. 172–179, 2019.